

Hit and Run ARMS: Adaptive Rejection Metropolis Sampling with Hit and Run Random Direction

Huaiye Zhang

Department of Statistics, Virginia Polytechnic
Institute and State University, Blacksburg, VA 24061, USA

Yuefeng Wu*

Department of Mathematics and Computer Science,
University of Missouri Saint Louis, Saint Louis, MO 63121, USA

Lulu Cheng and Inyoung Kim

Department of Statistics, Virginia Polytechnic
Institute and State University, Blacksburg, VA 24061, USA

March 10, 2015

Abstract

An algorithm for sampling from non-log-concave multivariate distributions is proposed, which improves the adaptive rejection Metropolis sampling (ARMS) algorithm by incorporating the hit and run sampling. It is not rare that the ARMS is trapped away from some subspace with significant probability in the support of the multivariate distribution. While the ARMS updates samples only in the directions that are parallel to dimensions, our proposed method, the hit and run ARMS (HARARMS), updates samples in arbitrary directions determined by the hit and run algorithm, which makes it almost not possible to be trapped in any isolated subspaces. The HARARMS performs the same as ARMS in a single dimension while more reliable in multidimensional spaces. Its performance is illustrated by a Bayesian free-knot spline regression example. We showed that it overcomes the well-known ‘lethargy’ property and decisively find the global optimal number and locations of the knots of the spline function.

Keywords: Adaptive Rejection Metropolis Sampling; Hit-and-Run Algorithm; Regression Splines; Free-knot Splines, Empirical Bayesian Method.

Classcode62G08, 62C12.

1 Introduction

Adaptive reject Metropolis sampling (ARMS) is a combination of adaptive rejection sampling (ARS) [1] and a Metropolis-Hastings sampling [2]. ARS was proposed for sampling from univariate log-concave conditional distributions. As an extension of ARS, ARMS [3] removes the log-concavity restriction on the simulated probability density functions. When the density to be sampled from is multidimensional, a straight-forward approach is to embed ARMS within a Gibbs sampler, provided that all one-dimensional conditional densities can be simulated by ARMS. This type of approaches have been used widely in various areas since the first paper on ARMS [3]. For some more examples, [4], [5],[6], [7], and [8] are all using ARMS in Gibbs sampler for multivariate distributions. However, if the probability density function is multimodal in a multidimensional space, Gibbs sampler can be easily trapped around some of the modes indefinitely, resulting in inefficient and even unreliable samples. In section 2, We showed that applying Gibbs sampler with ARMS algorithm to sample from a 2-dimensional distribution is trapped around 3 modes, while the target distribution has total 4 similar modes and is fairly smooth. Although, in practice, there are methods, such as using different proposal distributions, to prevent Gibbs or Metropolis-Hastings sampler being trapped in some subset of the support of the target distribution, whose probability is distinctly less than 1. These methods usually implemented by trial and error. They may still give impressions of convergence while some important aspects of the target distributions are missing. Hence, there is no guarantee from them that the samplers have sampled from all the subspaces around (almost) every local modes, especially sampling from multidimensional spaces and the locations of the modes are unknown priori.

While Metropolis-Hastings algorithm may be stuck in only part of the support of the distribution, applying ARMS to sampling from one-dimensional distribution does not generally have this problem, since it is an accept-reject method with an adaptive instrumental density function. In multidimensional situation, it is mainly due to the approach through Gibbs sampler that the ARMS can be trapped in some distinctly less than 1 probability subset of the support of the target distribution. Particularly, we believe that such being trapped disadvantage is due to that the Gibbs sampler updates the multidimensional samples in a fixed order and only in one dimension for each step. Note that Gibbs sampler does not necessarily sample in each dimension in every step, it could sample some dimension fewer times than the others and could sample in different order of the dimensions. Such alternative sampling scheme may relieve the being trapping issue at some level. However, such alternative schemes are not standard and need to be done by trial and fail. Therefore, again, this is not a very reliable way to solve the being trapped issue of ARMS in multi-dimensional distribution sampling.

We propose an algorithm, which incorporate hit and run method and ARMS, instead of embedding ARMS in Gibbs sampler, to sample from general multivariate distributions. Hit and Run algorithm was introduced by [9] and [10] Its property has been studied by [11] and [12]. Also, hit and run and Gibbs

samplers were compared by [13]. Although there's no theoretical result, the empirical study showed that the hit and run method estimators have less bias and standard errors than the corresponding Gibbs sampler estimators. The authors suggested that this is due to the less autocorrelations of the hit and run method than the Gibbs sampler. Plus, since hit and run method generates uniformly distributed directions, so it suffers the problem of being stuck in only part of the support of the distribution much less likely. Due to the difficulty of generating a signed distance for the hit and run method, grid, acceptance/rejection, or Metropolis methods have been used and the resulting algorithms have been studied. These methods more or less sacrifice efficiency. In this paper, ARMS algorithm is used to generate the signed distance. The resulting algorithm has not been proposed and studied explicitly in literature based on the authors' best knowledge. Combining the reliability of hit and run algorithm and the efficiency of ARMS, we can avoid the unreliable and inefficient issue due to the Gibbs sampler.

It is worth to mention that the hit and run is just one of the algorithms that deals with direction sampling. For example, adaptive direction sampling (ADS) was introduced in [14], where the authors mentioned the Gibbs sampler has much slower convergence rate in high dimensional situation compared to the ADS. Incorporating ADS and ARMS looks promising and worth to be studied and compared with the hit and run method with ARMS, but is out of the scope of this paper.

As an example of the application of HARARMS, it was applied to solve a free knots regression spline problem. Spline function is widely used in the analysis of two-dimensional data (y_i, x_i) . Early papers [15] [16] have addressed the importance of the role of splines in smoothing and regression analysis. Ruppert et al. A method of equally spaced knots was proposed in [17]. This knot placement method is simple and straightforward to implement. However, the nature of this method does not guarantee that knots are placed at all critical locations, at which the underlying regression function possesses sharp changes. The other type of methods is called curve-fitting with free-knot splines, where the number of knots and their locations are determined from the data. To discuss the full benefits of the spline approach, free-knot methods become an important and difficult problem.

Traditional methods [15] for the optimal knot selection is to add knots in intervals where the residuals are inadmissible large. Instead of parameters, the positions of the knots can be taken as the selection of functional type in an ordinary curve fitting problem [18]. The knots should be chosen as to correspond to the overall performance of the data fitting. A new knot is assigned until the sum of squared residuals gets to a minimum. The recent literature propose several approaches to automatic knot selection that needs to search all possible models. Many of them are based on stepwise regression ideas. For example, [19], [20], [21]. A stochastic search method was proposed in [22] for optimizing the knot locations by using a continuous genetic algorithm. Bayesian methods for fitting free-knot splines have been considered in some literatures [23], [24], [25], and [26]. These approaches employ continuous random search methodology

through the use of Monte Carlo Markov chain algorithms although the objective is not minimization of Equation. A review and comparison of some of these approaches is given by [27].

Although most of the automatic knot selection procedures mentioned here have exhibited good performance, they all face a nonlinear problem with a lethargy property [31]. The "lethargy" property is intrinsic to free-knot spline problems. In the other word, there may be local minima, saddle points, or even local maxima for the optimal knot placement. The "lethargy" typically cause many problems for the derivative-based optimization methodology and could lead to a poor estimation.

We show that the proposed HARARM algorithm has good properties for solving global optimization problems, in particular, finding the the numbers and locations of the free knot regression model efficiently and reliably. The overall procedure of the free-knot idea is investigated by simulated data sets.

The proposed HARARMS algorithm is described in detail in Section 2, where a brief description, which has more details than above, to the related algorithms are also given. Then by comparing and discussing the performances of sampling in multidimensional space by ARMS with Gibbs sampler and Hit and Run algorithm in Section 3, we show that HAEARMS is significantly better in multidimensional situation, especially there are multi-modes, which is the case in almost all real world problems. In the last section, the HARARMS is used to fit the data to the free knots regression spline, and the performance is assessed.

2 Algorithms

In the Bayesian context, the objectives of the modeling are usually posterior distributions of the unknown parameters. High dimensional distribution samples can be generated from the Gibbs sampling [28] straightforwardly for Bayesian inference. At each iteration of the Gibbs sampling, the parameter is updated by sampling a new value from its full conditional distribution. The full conditional distribution of a parameter is its distribution conditional on the data and on the current values of all the other parameters.

For the completeness, we introduce the adaptive rejection sampling(ARS) and the adaptive rejection Metropolis sampling(ARMS) first. We define x to be the random variable to be sampled.

The ARS extends the rejection sampling [29] by adaptively adjusting the sampling distribution. For a basic rejection sampling method, it requires that a sampling distribution $g(x)$ for a random variable x can be easily drawn and $Mg(x) \geq f(x)$ given a finite constant M . The rejection sample method is very useful for sampling a full conditional distribution since the integration of $f(x)$ is not needed. In practice, this algorithm involves an evaluation and potential rejection step after a sample is drawn from $g(x)$. For most of the time, it is a challenging task to pick up an appropriate sample distribution $g(x)$ and the constant M in order to reduce the ratio of the rejection.

The ARS [1] is proposed to use a sequence of sampling distributions $g_m(x)$

defined by piecewise linear functions $h_m(\mathbf{x})$:

$$h_m(\mathbf{x}) = \min[L_{l-1,l}(\mathbf{x}; S_m), L_{l+1,l+2}(\mathbf{x}; S_m)], \mathbf{x}_l \leq \mathbf{x} < \mathbf{x}_{l+1}, \quad (1)$$

where $S_m = \{\mathbf{x}_l, l = 0, \dots, m+1\}$ is a set of points in the support of f . $L_{lj}(\mathbf{x}; S_m)$ denotes the straight line through points $[\mathbf{x}_l, \ln f(\mathbf{x}_l)]$ and $[\mathbf{x}_j, \ln f(\mathbf{x}_j)]$. Defining $M_m = \int \exp(h_m(\mathbf{x}))d\mathbf{x}$, The sampling distribution is given by:

$$g_m(\mathbf{x}) = \frac{\exp(h_m(\mathbf{x}))}{M_m}. \quad (2)$$

Now the algorithm of ARS is derived as the following:

0. Initialize m and S_m .
1. Generate $\mathbf{x} \sim g_m(\mathbf{x}), U \sim \mathbb{U}_{[0,1]}$
2. If $U > \frac{f(\mathbf{x})}{\exp\{h_m(\mathbf{x})\}}$, update S_m to $S_{m+1} = S_m \cup \{\mathbf{x}\}$; otherwise, accept \mathbf{x} .

For univariate cases, the average of iterations of the ARS to accept one point depends on the initial S_m and the target distribution f . The biggest limitation of ARS is to require f to be log-concave since $h_m(\mathbf{x})$ needs to be an envelope of $\ln f(\mathbf{x})$. The ARMS [3] was proposed to deal with non-log-concave densities. The ARMS incorporates a Metropolis-Hastings algorithm step to ARS. The Metropolis-The Hastings algorithm is briefly covered as the following:

0. Initialize \mathbf{x}_s
1. Generate $\mathbf{x} \sim q(\mathbf{x}|\mathbf{x}_s), \sim \mathbb{U}_{[0,1]}$.
2. If $U > \min\{1, \frac{f(\mathbf{x})q(\mathbf{x}_s|\mathbf{x})}{f(\mathbf{x}_s)q(\mathbf{x}|\mathbf{x}_s)}\}$, $\mathbf{x}_{s+1} = \mathbf{x}$; otherwise accept \mathbf{x} .

The Metropolis-Hastings algorithm(MH) [2] improved the Metropolis algorithm [30] by generating samples from a proposal distribution $q(\mathbf{x}|\mathbf{x}')$. The performance of the MH algorithm depends on the quality of the proposal distribution. Similar to other MCMC algorithms, samples from the MH algorithm may stay in a local maximum of f due to an unsuitable proposal distribution.

To carry out the ARMS algorithm [3], let $S_m = \{\mathbf{x}_l; l = 0, \dots, m+1\}$ denote a current set of points in ascending order, For $1 \leq l \leq m$, let $L_{lj}(\mathbf{x}; S_m)$ denote the straight line through points $[\mathbf{x}_l, \ln f(\mathbf{x}_l)]$ and $[\mathbf{x}_j, \ln f(\mathbf{x}_j)]$. A piecewise linear function $h_{l,l+1}(\mathbf{x}) = \max[L_{l,l+1}(\mathbf{x}), \min\{L_{l-1,l}(\mathbf{x}), L_{l+1,l+2}(\mathbf{x})\}]$, where $\mathbf{x} \in (\mathbf{x}_l, \mathbf{x}_{l+1})$. $h_m(\mathbf{x}) = \{h_{l,l+1}(\mathbf{x}); l = 1, \dots, m-1\}$. Let $h_{0,1}(\mathbf{x}) = L_{0,1}(\mathbf{x})$ and $h_{m,m+1}(\mathbf{x}) = L_{m,m+1}(\mathbf{x})$ define the first and the last piecewise linear functions, respectively. The sampling distribution is then $g_m(\mathbf{x}) = \exp\{h_m(\mathbf{x})\}/M_m$, where $M_m = \int \exp\{h_m(\mathbf{x})\}d\mathbf{x}$. We further define \mathbf{x}_{cur} and \mathbf{x}_{new} as the current value and new sample from $f(\mathbf{x})$. ARMS starts from $h_{0,1}(\mathbf{x}) = L_{0,1}(\mathbf{x})$ and construct $h_{i,i+1}(\mathbf{x})$; we include a new point and relabel points; we then reconstruct $h_{l,l+1}(\mathbf{x})$. For multivariate distribution, we reserve the bold font \mathbf{x} for the

objective random variable vector. Let k denote the dimension of the variable and $k = 1, \dots, K$. Assuming $f(\mathbf{x}_{(k)}|\mathbf{x}_{(-k)})$ is a target univariate distribution to be selected, where $\mathbf{x}_{(-k)}$ is the parameters except $\mathbf{x}_{(k)}$. For ease of notation we write $f(\mathbf{x})$ below instead of $f(\mathbf{x}_{(k)}|\mathbf{x}_{(-k)})$. The algorithm of ARMS embedded in Gibbs sampler for multivariate distribution is described as the following:

For $k = 1$ to K , let $f(\mathbf{x})$ denote $f(\mathbf{x}_{(k)}|\mathbf{x}_{(-k)})$ in the following steps. Do

1. Simulate \mathbf{x}_A from $g_m(\mathbf{x})$ and $U \sim \mathbb{U}_{[0,1]}$ until

$$U \leq \frac{f(\mathbf{x}_A)}{\exp\{h_m(\mathbf{x}_A)\}}.$$

2. Generate $U \sim \mathbb{U}_{[0,1]}$ and take

$$\mathbf{x}_{(g)} = \begin{cases} \mathbf{x}_A & U < \min\left\{1, \frac{f(\mathbf{x}_A) \min[f(\mathbf{x}_{cur}), \exp(h_m(\mathbf{x}_{cur}))]}{f(\mathbf{x}_{cur}), \min[f(\mathbf{x}_A) \exp(h_m(\mathbf{x}_A))]} \right\}, \\ \mathbf{x}_{cur} & otherwise. \end{cases}$$

The Hit-and-Run algorithm can be thought of as random-direction Gibbs: in each step of the hit-and-run algorithm, instead of updating \mathbf{x} along one of the dimensions, we update it along a randomly generated direction that is not necessarily parallel to any dimension. More precisely, the sampler is defined in two steps: first, choose a direction d from some positive density on the unit sphere $d^T d = 1$. Then, similar to Gibbs, sample the new point \mathbf{x}_{new} along the line specified by d and the distance by z , where z is from the marginal one-dimensional density on the line that specified by d . The ARMS with Hit-and-Run (HARARMS) algorithm is as the following:

0. Initiate \mathbf{x}_s , generate $d_g = u_g / \sqrt{u_1^2 + \dots + u_G^2}$, $u_g \sim \mathbb{U}_{[0,1]}$, and set $f^*(z) = f(\mathbf{x}_s + d^T z)$, initiate M and S_m for $f^*(z)$.

1. Simulate z_A from $g_m(z)$ and $U \sim \mathbb{U}_{[0,1]}$ until

$$U \leq \frac{f(z_A)}{\exp\{h_m(z_A)\}}.$$

2. Generate $U \sim \mathbb{U}_{[0,1]}$ and take

$$\mathbf{x}_s = \begin{cases} \mathbf{x}_{s-1} + d^T z_A & U < \min\left\{1, \frac{f(z_A) \min[f(z_{cur}), \exp(h_m(z_{cur}))]}{f(z_{cur}), \min[f(z_A) \exp(h_m(z_A))]} \right\}, \\ \mathbf{x}_{s-1} + d^T z_{cur} & otherwise. \end{cases}$$

3 Comparing sampling performances

The ARMS with Hit-and-Run reduces a multiple dimensional question into one dimensional ARMS sampling. It breaks the procedure into two steps: (1). pick up a random direction d ; (2). implement a one dimensional ARMS for the random variable z . An important advantage of ARMS with Hit-and-Run over regular multiple dimensional ARMS is that it is much more likely to reach the isolated local areas by evaluating one dimensional ARMS in a random direction searching, while regular multiple dimensional ARMS only searches the space in the direction that is parallel to one of dimensions in each updating step. For example, in a two dimensional case, the Gibbs sampler sampling the new points either in the vertical or the horizontal direction from the current point. The following example shows that even for sampling from a mixture of 4 bivariate normal distribution with similar mixing probability mass and variances, the Gibbs sampler with ARMS can be trapped around only 3 of the 4 modes and totally missing the forth one. This is mainly due to that the forth mode is hardly reached by searching in the 2 dimensional space along the directions only parallel to the axes in each step.

Assuming that we have a two dimensional random variable, $\mathbf{x} = \{x_{(1)}, x_{(2)}\}^T$, and the the objective sampling distribution is $f(\mathbf{x})$ as below:

$$f(\mathbf{x}) = \sum_{i=1}^4 p_i N(\mathbf{x} | \mu_i, \Sigma). \quad (3)$$

In this example, we set $\mu_1 = (5, -5)^T$, $\mu_2 = (5, 5)^T$, $\mu_3 = (-5, 5)^T$, and $\mu_4 = (13, 13)^T$. $\Sigma = \text{Diag}(0.5, 0; 0, 0.5)$, $p_1 = 0.2$, $p_2 = 0.3$, $p_3 = 0.2$ and $p_4 = 1 - p_1 - p_2 - p_3$.

Sampling \mathbf{x} from $f(\mathbf{x})$ is an easy task if we know that $f(\mathbf{x})$ is a mixture of normal distributions. We can pick one sampling distribution among the four in terms of their probabilities p_i , and then sample \mathbf{x} from the selected normal distribution. Consider $f(\mathbf{x})$ as if it did not have the explicit and easy form of mixture to sampling form, and treat it as a general distribution, which is the case as sampling from general complex functions. Then we use both the Gibbs sampler with ARMS embedded and the HARARMS to sample \mathbf{x} from the $f(\mathbf{x})$, and compare their performances. After $s=10000$ iteration, we generate the sampling of \mathbf{x} from both Gibbs sampler with ARMS embedded and HARARMS. To make a reference, contour plots are made by the grid search on $x_{(1)}$ and $x_{(2)}$, which refer as x_1 and x_2 in the Figure 1.

The comparison of the performances of the two algorithms is also summarized by the following table:

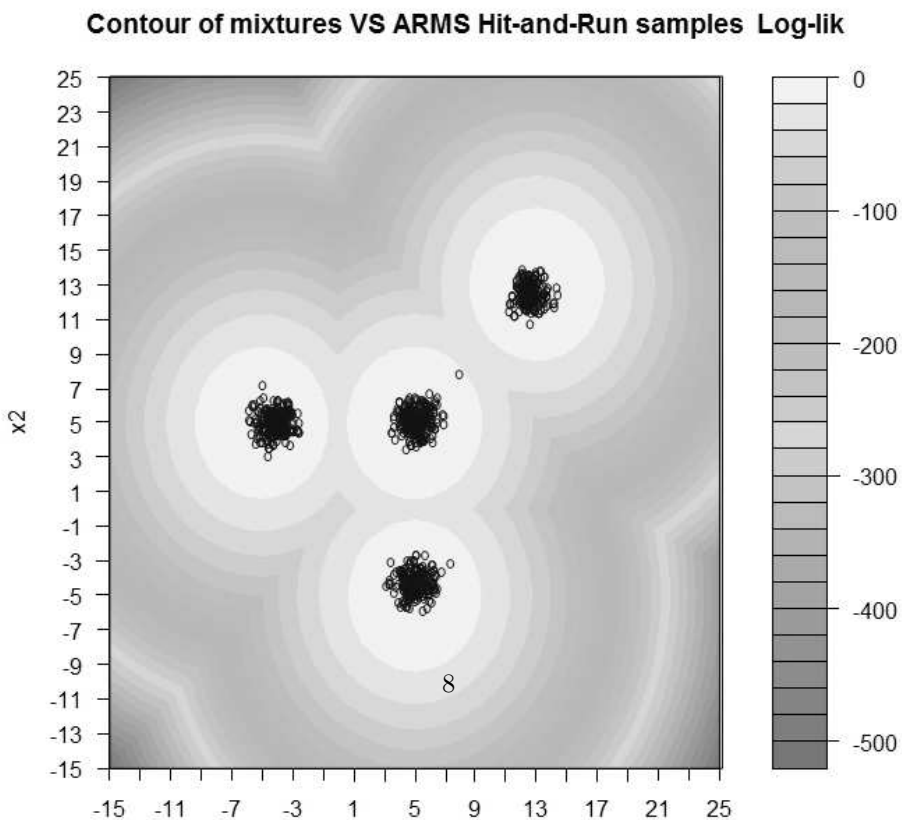
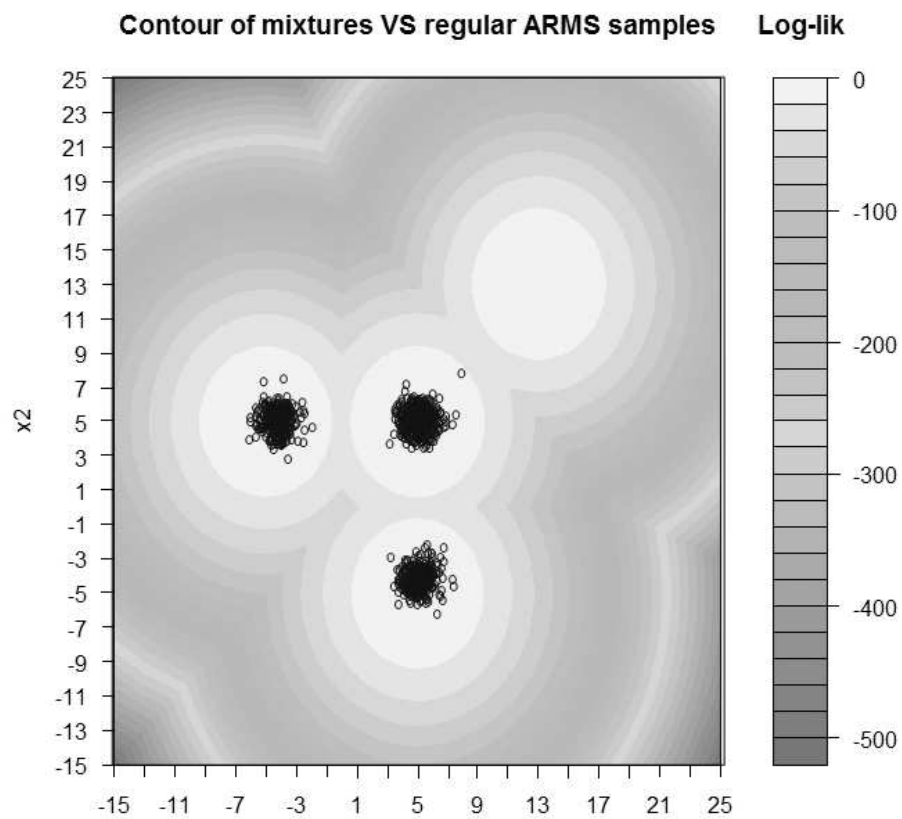


Figure 1: Top: shows that Gibbs-ARMS is trapped around only 3 of the 4 modes. Bottom: shows that HARARMS samples from around every modes.

Table 1: Comparing ARMS performances

Parameter	Actual			HARARMS			ARMS with Gibbs		
	μ_1	μ_2	p	μ_1	μ_2	p	μ_1	μ_2	p
(Comp 1)	5	-5	0.20	5.00	-4.98	.20	4.99	-5.03	.29
(Comp 2)	5	5	0.30	4.99	5.02	.30	4.99	5.02	.42
(Comp 3)	-5	5	0.30	-4.97	4.99	.22	-4.98	4.99	.29
(Comp 4)	13	13	0.20	12.98	13.02	.28			

The HARARMS estimates all the parameters well, while the Gibbs sampler misses one of the modes and gives severely biased estimation on the mixing probability.

4 Free Knots Regression splines

Now we apply the HARARMS to a free knot spline model. Assuming that we would like to model a dependent variable, y_i , using one independent variable, x_i , where $i = 1, \dots, n$. Starting with the straight line regression model:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i,$$

where $\epsilon_i \sim N(0, \sigma^2)$. Denote $x = (x_1, \dots, x_n)^T$ and $y = (y_1, \dots, y_n)^T$. In the framework of spline regression, the basis functions for fitting such simple linear regression are defined as:

$$X = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}.$$

To handle nonlinear structure, regression models consisting of several differently sloped lines are developed. One method is to introduce a basis function that is zero to the left of κ and then is a positive value from κ onward. A mathematical way of expressing this truncated basis function is

$$(x_i - \kappa)_+,$$

where, for any number u , u_+ is equal to u if u is positive and is equal to 0 otherwise. Therefore, the regression model with multiple truncated basis functions is

$$y_i = \beta_0 + \beta_1 x_i + \sum_{k=1}^K b_k (x_i - \kappa_k)_+ + \epsilon_i, \quad i = 1, \dots, n. \quad (4)$$

Now the corresponding basis function matrix is:

$$X = \begin{bmatrix} 1 & x_1 & (x_1 - \kappa_1)_+ & \cdots & (x_1 - \kappa_K)_+ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & (x_n - \kappa_1)_+ & \cdots & (x_n - \kappa_K)_+ \end{bmatrix}.$$

In (4) the value of κ corresponding to the function $(x_i - \kappa)_+$ usually refers to

a knot. A set of such functions is called a linear spline basis. Equation (4) is called a spline model with a linear spline basis. Assume that the parameters β_0, β_1 , as well as $\mathbf{b} = (b_1, \dots, b_k, \dots, b_K)$ are unknown while κ 's in matrix \mathbf{X} is known, the model can be easily solved by the multiple linear regression.

However, if the number and locations of the knots, the κ 's, are unknown, the model will be not so easy to handle, and the choice of them is a critical problems. Wold [18] describes the choice of knot positions as the selection of functional type. The knots should be chosen in terms of the overall performance of the data fitting. Simplified methods ([15], [17]) works well when the number of knots is large (about 30-40). The extra caution is necessary in using the large number of knots, since the great flexibility of splines can make overfitting the data. Ideally we should put as few knots as possible. The knots should be placed at all critical locations, at which the underlying regression function possesses sharp changes.

The difficulty in detecting the optimal number and location of knots comes from the "lethargy" property ([31]) intrinsic to free-knot splines as discussed in Section 1. Such local maxima problem for free-knot splines method is rarely discussed, and it actually is the big barrier for designing an attractive free-knot algorithm. We show the local maxima problem for model (4) in detail as the following:

To find the optimal number and locations of the knots, we define the loss function by the sum of squared errors, just the same as the classical method, which minimizes the loss function and finds the optimal. Let $\mathbf{B} = (\beta_0, \beta_1, \mathbf{b})^T$. Based on (4) and assumption that the error is normally distributed, the full likelihood is:

$$p(y_i|\kappa, \mathbf{B}, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp[-\frac{1}{2}\sigma^2(y_i - \mathbf{x}_i(\kappa)\mathbf{B})^2] \quad (5)$$

In (5), both of knot locations κ and the coefficients \mathbf{B} and σ^2 need to be optimized. Given the number and locations of knots, the optimization in terms of the coefficients, \mathbf{B} and σ^2 , can be obtained by standard linear least-squares solution. In the other word, we have the residual sum of squares for a given knots' locations by substituting \mathbf{B} and σ^2 with the ordinary least squares solution $\hat{\mathbf{B}}$ and an unbiased estimation $\hat{\sigma}^2$. Consider an empirical Bayesian method:

$$p(y_i|\kappa) \propto \frac{1}{\sqrt{2\pi}\hat{\sigma}^2} \exp[-\frac{1}{2}\hat{\sigma}^2(y_i - \mathbf{X}_i(\kappa)\hat{\mathbf{B}})^2] \quad (6)$$

where $\hat{\mathbf{B}}$ is an empirical solution, i.e. $\hat{\mathbf{B}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$. $\hat{\sigma}^2 = \sum_i (y_i - \mathbf{X}_i\hat{\mathbf{B}})/(n - q - 1)$, where q is the number of unknown coefficients of (4). It is now well known that the likelihood function corresponding to (6) tends to have multiple maxima and exhibits a lethargy property near extremes that can lead to premature convergence at nonoptimal knot locations.

To show the multiple maxima of the likelihood function, we give a noninfor-

mative prior to κ . Then the posterior distribution of κ is:

$$p(\kappa|y) \propto \prod_{i=1}^n p(y_i|\kappa). \quad (7)$$

Therefore, the Log-likelihood function for κ is

$$l(\kappa, y) = \sum_{i=1}^n \log(p(y_i|\kappa)). \quad (8)$$

Now we generate example data sets, and show the existence of the local maxima in terms of the location of the knots. Generate a simulated Dataset 1, (x_i, y_i) , from (4) with the parameters values as summarized in the following Table 2, and the generated data is plotted in Figure2.

Table 2: Simulated Dataset 1

Type	Value
Knot location, κ	$\kappa=(200, 300, 400, 500, 700, 900)$
Coefficients, \mathbf{B}	$\beta_0 = -0.5, \beta_1 = -0.5$, and $\mathbf{b} = (0.5, 1.0, -2.0, 2.5, -3.0, 3.5)$
Error, ϵ_i	$\epsilon_i = N(0, 30^2)$
Independent variable, x	$x = (1, 2, \dots, 1000)^T$
Dependent variable, y	Sampled based on (4)

Similar to the setting of dataset 1, let κ equal to (700) or (700,500), and then plot the curve of the likelihood functions (8) versus the location of κ in Figure 3 correspondingly. Using one and two dimensional κ is to make the graph easy to read, without loss of generality. The multiple maxima are clear in the graph. Specifically, for $K = 1$, the largest value of the Log-likelihood (-5630.89) occurs with knot location at 730; a local maxima occurs at 160, and the Log-likelihood is -5698.99. For $K = 2$, Let both of κ_1 and κ_2 have values from 100 to 990 with increments of 10. The value of the Log-likelihood is then plotted for each combination of the two knot locations on the right side of Figure 3. You can see that the largest value of the Log-likelihood (-5339.10) occurs with knot location at a splines function with $\kappa_1 = 670$ and $\kappa_2 = 520$; A local maxima Log-likelihood (-5406.38) appears at a splines function with $\kappa_1 = 700$ and $\kappa_2 = 270$; The other local maxima Log-likelihood (-5535.50) appears at the combination of $\kappa_1 = 90$ and $\kappa_2 = 770$. Through this visualization, It is clear that three local Log-likelihood maximums exist within the splines function with two knots. This indicates that the log-likelihood function for the general free knot spline with unknown number of knots should be multimodal in the high dimensional space. we can see that the location of a global maximum can therefore be assured only through a global grid search, which unfortunately becomes computationally infeasible quickly as the dimension increases. As we discussed above, for sampling from such target functions, the HARARMS will be better than Gibbs sampler.

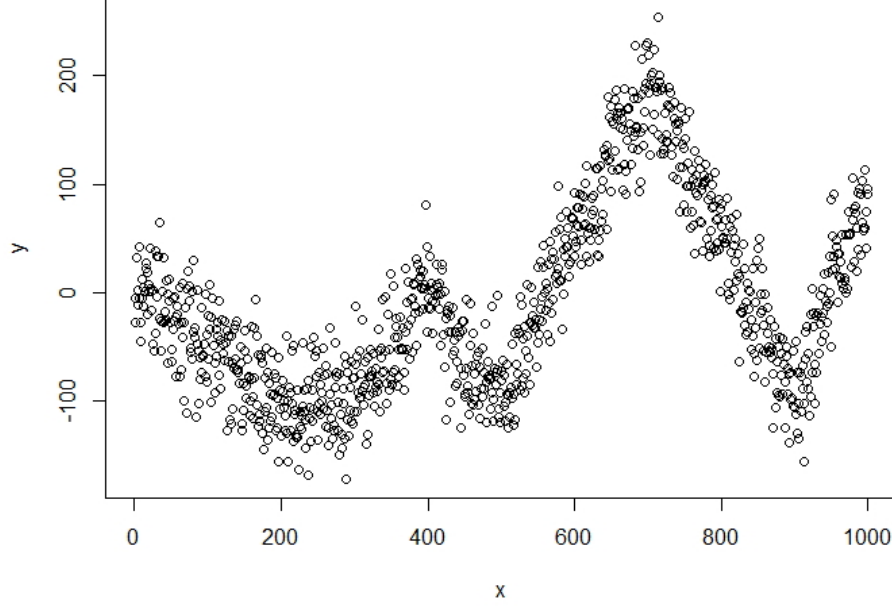


Figure 2: y is generated from (4) with defined knot locations and parameters values. $x = (1, 2, \dots, 1000)^T$

Table 3 shows another setting for generating a quadratic spline function with 5 knots. The data generated from the this setting is named as dataset 2.

Table 3: Simulated Dataset 2 - for Quadratic Spline base

Type	Value
Knot location, κ	$\kappa = (0.2, 0.4, 0.5, 0.7, 0.9)$
Coefficients, \mathbf{B}	$\beta_0 = -0.5, \beta_1 = 0.5, \beta_2 = -0.5$, and $\mathbf{b} = (1, -3, 5, -7, 15)$
Error, ϵ_i	$\epsilon_i = N(0, 0.3^2)$
Independent variable, x	$x = (1, 2, \dots, 1000)^T$
Dependent variable, y	Sampled based on (4)

The following 2 tables and 2 figures are the result of using HARARMS to find fit the free knot spline model to dataset 1 and dataset 2 correspondingly. Since the number of the knots in the true model is unknown, we fitted the models with $1, 2, \dots, 10$ knots correspondingly. As the number of the knots increases, the maximum of the likelihood increases, even when the number is greater than the true number. However, by comparing the increasing among the different numbers of knots in the model, we can easily identify the true model. When

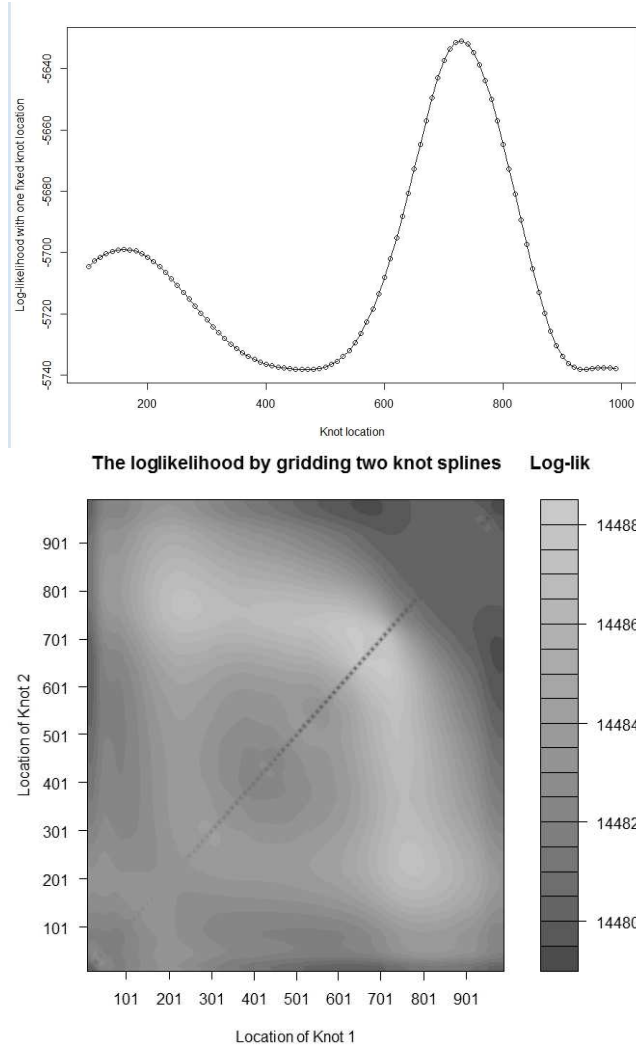


Figure 3: Top: Gridding the knot location with one knot splines, plot the log likelihood of (6) to demonstrate the multiple local maximum problem. Bottom: Gridding the knot location with two knot splines, plot the log likelihood of (6) to demonstrate the multiple local maximum problem.

the true number is reached, adding more knots will only increase the maximum of the log-likelihood function marginally. Of course, based on this finding, using AIC or BIC as criteria can decisively and correctly get the model selection done. The estimation of the location of the knots when the number of the knots in the model is selected correctly, is quite accurate, as we can see in the following tables.

Table 4: Linear regression splines

Case	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8	κ_9	κ_{10}	Log-likelihood
Actual	200	300	400	500	700	900					
1-knot	727										-5631.07
5%	713										-5633.01
95%	740										-5630.87
2-knot	518	671									-5339.78
5%	508	661									-5342.61
95%	530	679									-5339.11
3-knot	519	698	899								-5040.63
5%	511	694	894								-5043.65
95%	526	702	904								-5039.79
4-knot	185	540	698	899							-4925.23
5%	172	532	694	894							-4928.52
95%	198	548	701	903							-4923.74
5-knot	245	404	503	697	899						-4802.55
5%	232	398	498	694	896						-4805.52
95%	258	409	507	700	903						-4800.69
6-knot	206	286	400	503	698	899					-4796.77
5%	168	255	394	496	695	896					-4800.75
95%	223	309	407	507	700	903					-4794.36
7-knot	211	287	400	503	697	899	969				-4796.57
5%	188	264	394	498	694	884	899				-4799.86
95%	234	314	406	507	700	903	987				-4794.46
8-knot	207	227	280	359	397	503	697	899			-4796.20
5%	155	185	230	313	383	498	694	895			-4799.98
95%	224	241	309	398	410	508	700	903			-4794.09
9-knot	30	213	272	312	396	462	504	698	899		-4795.85
5%	13	118	210	280	376	411	498	694	896		-4799.03
95%	86	231	311	368	405	507	546	701	903		-4793.67
10-knot	27	179	218	277	313	397	474	505	697	899	-4795.03
5%	12	147	182	232	271	370	393	499	695	895	-4797.91
95%	61	222	279	305	365	407	508	550	700	903	-4793.02

The generated data is following a linear spline function with 6 knots at 200, 300, 400, 500, 700, and 900. The n -th big row shows the estimation for the location of the knots, corresponding to the fitted model has n knots. Also, the 90% credible intervals to these estimation are reported. In the last column, the value of the log-likelihood function corresponding to the estimated values and the boundaries of the intervals are listed. Not surprisingly, the log-likelihood function value of the estimations increases as the number of the knots increases. However, when the number of the knots in the model reaches the truth, the increasing in log-likelihood could be neglected, compared to the increasing before the number of knots in the model reaches the true number.

Table 5: Quadratic regression splines

Case	κ_1	κ_2	κ_3	κ_4	κ_5	κ_6	κ_7	κ_8	κ_9	κ_{10}	Log-likelihood
Actual	0.2	0.4	0.5	0.7	0.9						
1-knot	0.77										-1853.74
5%	0.77										-1854.35
95%	0.77										-1853.35
2-knot	0.59	0.64									-1616.39
5%	0.57	0.62									-1618.44
95%	0.61	0.65									-1615.81
3-knot	0.58	0.69	0.90								-1276.31
5%	0.57	0.68	0.90								-1279.13
95%	0.59	0.70	0.91								-1275.32
4-knot	0.45	0.48	0.70	0.90							-1205.32
5%	0.43	0.47	0.69	0.90							-1207.94
95%	0.47	0.50	0.71	0.91							-1203.96
5-knot	0.19	0.40	0.49	0.70	0.90						-1184.38
5%	0.15	0.36	0.47	0.69	0.90						-1187.97
95%	0.24	0.44	0.51	0.71	0.91						-1182.81
6-knot	0.19	0.41	0.49	0.69	0.72	0.90					-1184.09
5%	0.14	0.38	0.47	0.61	0.69	0.90					-1187.57
95%	0.23	0.44	0.51	0.71	0.79	0.91					-1182.59
7-knot	0.20	0.39	0.47	0.51	0.70	0.86	0.90				-1183.19
5%	0.15	0.27	0.41	0.47	0.67	0.69	0.89				-1186.42
95%	0.26	0.44	0.51	0.70	0.75	0.91	0.96				-1181.28
8-knot	0.10	0.20	0.41	0.48	0.55	0.68	0.74	0.90			-1182.46
5%	0.01	0.02	0.15	0.39	0.48	0.51	0.68	0.89			-1185.40
95%	0.25	0.40	0.47	0.51	0.71	0.74	0.91	0.96			-1180.74
9-knot	0.19	0.31	0.43	0.48	0.60	0.68	0.72	0.87	0.91		-1182.08
5%	0.04	0.14	0.29	0.39	0.47	0.52	0.64	0.69	0.89		-1185.53
95%	0.26	0.43	0.51	0.60	0.69	0.72	0.90	0.92	0.98		-1180.18
10-knot	0.06	0.18	0.28	0.40	0.47	0.52	0.64	0.69	0.77	0.90	-1181.09
5%	0.02	0.06	0.15	0.29	0.36	0.44	0.48	0.63	0.68	0.89	-1184.02
95%	0.24	0.33	0.42	0.50	0.54	0.66	0.71	0.78	0.91	0.94	-1179.21

The generated data is following a quadratic spline function with 6 knots at 0.2, 0.4, 0.5, 0.7 and 0.9. The n -th big row shows the estimation for the location of the knots, corresponding to the fitted model has n knots. Also, the 90% credible intervals to these estimation are reported. In the last column, the value of the log-likelihood function corresponding to the estimated values and the boundaries of the intervals are listed. Not surprisingly, the log-likelihood function value of the estimations increases as the number of the knots increases. However, when the number of the knots in the model reaches the truth, the increasing in log-likelihood could be neglected, compared to the increasing before the number of knots in the model reaches the true number.

References

- [1] Gilks WR, Wild P. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 1992;41:337-348.
- [2] Hastings WK. Monte Carlo sampling methods using Markove chains and their applications. *Biometrika*, (1970).;57:97-109.
- [3] Gilks WR, Best NG, Tan KKC. Adaptive rejection Metropolis sampling. *Applied Statistics*, 1995;44:455-472
- [4] Meyer R, Millar BR. Bayesian stock assessment using a statespace implementation of the delay difference model. *Canadian Journal of Fisheries and Aquatic Sciences*, 1999; 56(1):37-52.
- [5] Meyer R, Christensen N. Bayesian reconstruction of chaotic dynamical systems. *Physical Review E*, 2000; 62(3):3535.
- [6] Putter H, Heisterkamp SH, Lange JMA, de Wolf F. A Bayesian approach to parameter estimation in HIV dynamical models. *Statistics in Medicine*, 2002;21:2199-2214.
- [7] Miranda MF, Zhu H, Ibrahim JG. Bayesian spatial transformation models with applications in neuroimaging data. *Biometrics*, 2013;69(4):1074-1083.
- [8] Yelland PM, Dong X. Forecasting demand for fashion goods: a hierarchical Bayesian approach. *Intelligent Fashion Forecasting Systems: Models and Applications*. Springer Berlin Heidelberg, 2014;71-94.
- [9] Schmeiser BW. Random variate generation: a survey. *Simulation with discrete models: state of the art view*, T.I. Oren, C.M.Shub, P.f.Roth(eds), IEEE, 1980;:79-104.
- [10] Rubinstein RY. Generating random vectors uniformly distributed inside and on the surface of different regions. *European Journal of Operations Research*, 1982; 10:205-209.
- [11] Andersen HC, Diaconis P. Hit and run as a unifying device. *Yournal de la société française de statistique*, 2007;148(5):5-28.
- [12] Lovász, László, and Santosh Vempala. Hit-and-run is fast and fun. preprint, Microsoft Research, 2003.
- [13] Chen MH, Schmeiser B. Performance of the Gibbs, hit-and-run, and Metropolis samplers. *Journal of computational and graphical statistics*, 1993;2(3):251-272.
- [14] Roberts GO, Gilks WR. Convergence of adaptive direction sampling. *Journal of multivariate analysis*, 1994;49(2):287-298.

- [15] Rice JR. The approximation of functions, Vol. 2., Addison-Wesley, Reading, Mass; 1969.
- [16] Smith PL. Splines as a useful and convenient statistical tool. *The American Statistician*, 1979;33(2):57-62.
- [17] Ruppert D, Wand M, Carroll R. *Semiparametric Regression*, Cambridge University Press, Cambridge, 2003.
- [18] Wold S. Spline functions in data analysis. *Technometrics*, 1974;16(1):1-11.
- [19] Smith PL. Hypothesis testing in B-spline regression. *Communications in Statistics, Part B*, 1982;11:143-57.
- [20] Friedman JH, Silverman BW. Flexible parsimonious smoothing and additive modeling. *Technometrics*, 1989;31:3-21.
- [21] Stone CJ, Hansen MH, Kooperberg C, Truong YK. Polynomial splines and their tensor products in extended linear modeling. *Annals of Statistics*, 1997;25:1371-1425.
- [22] Spiriti S, Eubank R, Smith PW, Young D. Knot selection for least-squares and penalized splines. *Journal of Statistical Computation and Simulation*, 2013;83:1020-1036.
- [23] Smith Mc Kohn R. Nonparametric regression using Bayesian variable selection. *Journal of Econometrics*, 1996;75(2):317-343.
- [24] Denison, DGT, Mallick, BK, and Smith, AFM. Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 1997;60(2), 333-350.
- [25] DiMatteo I, Genovese CR, Kass RE. Bayesian curve fitting with free-knot splines. *Biometrika*, 2001; 88(4):1055-1071.
- [26] Lindstrom, MJ. Bayesian estimation of free knot splines using reversible jumps. *Computational statistics & data analysis*, 2002;41(2):255-269.
- [27] Wand MP. A comparison of regression spline smoothing procedures. *Computational Statistics*, 2000;15:443-62.
- [28] Gelfand AE, Smith AF. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 1990;85(410):398-409.
- [29] Ripley, B. *Stochastic Simulation*. New York, Wiley; 1987.
- [30] Metropolis N, Rosenblth AW, Rosenbluth MN, Teller AH, Teller E. Equations of state calculations by fast computing machines *J. Chem Phys.*, 1953;21:1087-1092.
- [31] Jupp DL. Approximation to data by splines with free knots. *SIAM Journal on Numerical Analysis*, 1978;15(2):328-343.